Rule-Based Automated Price Negotiation: an Overview and an Experiment

Costin Bădică[1], Maria Ganzha[2], and Marcin Paprzycki[3]

[1] University of Craiova, Software Engineering Department
Bvd.Decebal 107, Craiova, RO-200440, Romania
badica_costin@software.ucv.ro
[2] Elbląg University of Humanities and Economy, ul Lotnicza 2, 82-300 Elbląg, Poland
ganzha@euh-e.edu.pl
[3] Computer Science Institute, Warsaw School of Social Psychology, 03-815 Warsaw, Poland
Marcin.Paprzycki@swps.edu.pl

Abstract. Recently, the idea of automating e-commerce transactions attracted a lot of interest. Multi-agent systems are one of promising software technologies for achieving this goal. This note discusses rule-based approaches to automated negotiation and presents some experimental results based on our own implementation of a rule-based price negotiation mechanism in a model e-commerce multi-agent system. The experimental scenario considers multiple buyer agents involved in multiple English auctions that are performed in parallel.

1 Introduction

During last few years, interest in e-commerce has shifted from simple Web presence of businesses to advanced use of e-commerce technologies. These technologies are used to support the growth of the business itself, by improving its efficiency and profitability. Therefore, the idea of automating e-commerce transactions attracted a lot of research.

Currently, e-commerce systems are based on humans to make the most important decisions in various activities along the lifeline of an e-commerce transaction. Software agents are often claimed to be one of the best technologies for automating e-commerce processes. In this context, we have set up a project to contribute to bridging the gap between these two scenarios (real and possible). Our project has two main goals: i) to build a large-scale implementation approximating an e-commerce environment; ii) to develop a tool that can be used for modeling various e-commerce scenarios.

E-commerce utilizes (to various degrees) digital technologies to mediate commercial transactions. As a part of our project we have conceptualized a commercial transaction as consisting of four phases: i) *pre-contractual phase* including activities like need identification, product brokering, merchant brokering, and matchmaking; ii) *negotiation* where participants negotiate according to the rules of the market mechanism and using their private negotiation strategies; iii) *contract execution* including activities like: order submission, logistics, and payment; iv) *post-contractual phase* that includes activities like collecting managerial information and product or service evaluation. Focus of this paper is on negotiation phase in a multi-agent e-commerce system that we been developing and implementing ([7], [1]). As part of this work we are interested in endowing our agents with flexibility necessary to engage in unknown in advance forms of negotiations using rule-based approaches. Therefore we start with a brief overview of background work on rule representations in automated negotiation. Then we outline the design of our system that uses rules for enforcing specific negotiation mechanisms, together with a sample experimental scenario. Finally, we discuss experimental results obtained using our implementation (see [2] for implementation details).

2 Overview of Rule-Based Negotiation

Rules have been indicated as a promising technique for formalizing multi-agent negotiations ([3, 4, 6, 11, 13, 16, 2]). Note that when designing systems for automated negotiations one should distinguish between *negotiation protocols* (or *mechanisms*) that define ”rules of encounter” between participants and *negotiation strategies* that define behaviors aiming at achieving a desired outcome. Thus far multiple rule representations were proposed for both negotiation mechanisms ([3, 14, 11, 16]) and strategies ([6, 13, 4]).

In our work we follow a rule-based framework for enforcing specific negotiation mechanisms inspired by [3]. Its authors sketched a complete framework for implementing portable agent negotiations that comprises: (1) negotiation infrastructure, (2) generic negotiation protocol and (3) taxonomy of declarative rules. The *negotiation infrastructure* defines roles of *negotiation participants* and of a *host*. Participants exchange proposals within a negotiation locale managed by the host. The *generic negotiation protocol* defines three phases of a negotiation: (1) admission, (2) exchange of proposals, and (3) formation of an agreement, in terms of how and when messages should be exchanged between the host and participants. *Negotiation rules* are used for enforcing the negotiation mechanism. Rules are organized into taxonomy: rules for participants admission to negotiations, rules for checking validity of proposals, rules for protocol enforcement, rules for updating negotiation status and informing participants, rules for agreement formation and rules for controlling negotiation termination.

The proposal for formalizing negotiations introduced in [14] goes beyond the framework of [3]. Its authors suggest use of an ontology for expressing negotiation protocols. Whenever an agent is admitted to negotiation it also obtains a specification of the negotiation rules in terms of the shared ontology. In some sense, the negotiation template used by our implementation (see [2]) is a ”simplified” negotiation ontology and the participants must be able to ”understand” parameters defined in the template. This approach is exemplified with a sample scenario, but [14] contains neither implementation details, nor experimental results.

In [16] a mathematical characterization of auction rules for parameterizing the auction design space is introduced. The proposed parametrization is organized along three axes: i) *bidding rules* – state when bids may be posted, updated or withdrawn; ii) *clearing policy* – states how the auction commands resource allocation (including auctioned items and money) between auction participants (this corresponds roughly to agreement making in our approach); iii) *information revelation policy* – states how and what intermediate auction information is supplied to participating agents.

In [11] an implementation of a new rule-based language for expressing auction mechanisms – AB3D scripting language — is reported. The design and implementation of AB3D were primarily influenced by the parametrization of auction design space defined in [16] and the previous experiences with the Michigan Internet AuctionBot ([15]). According to [11], A3BD allows the initialization of auction parameters, the definition of rules for triggering auction events, declaration of user variables and definition of rules for controlling bid admissibility.

A formal executable approach for defining strategy of agents participating in negotiations using defeasible logic programs is reported in [8] and [6]. This approach was demonstrated using English auctions and bargaining with multiple parties by indicating sets of rules for describing strategies of participating agents. However, no implementation results were reported. It is also interesting to note that despite the fact that paper [8] claims to address both protocol and strategy representation in defeasible logic, finally only a simplistic example of a protocol for two bidding rounds auction is described, while more detailed examples are provided only for strategy representation.

In [13] a preliminary implementation of a system of agents that negotiate using strategies expressed in defeasible logic is described. The implementation is demonstrated with a bargaining scenario involving one buyer and one seller agent. The buyer strategy is defined by a defeasible logic program. Note that the implementation reported in [13] builds on the architecture of negotiating agents previously introduced in [6].

The CONSENSUS system that enables agents to engage in combined negotiations was presented in [4]. CONSENSUS allows agents to negotiate different complementary items on separate servers on behalf of human users. Each CONSENSUS agent uses rules partitioned into: i) *basic rules* that determine the negotiation protocol, ii) *strategy rules* that determine the negotiation strategy and iii) *coordination rules* that determine knowledge for assuring that either all of the complementary items or none are purchased. Note that in CONSENSUS the rule-based approach is taken beyond mechanism and strategy representation to capture also coordination knowledge.

3 System Description and Experiment

Conceptual Architecture. Our system acts as a distributed marketplace in which agents perform functions typically observed in e-commerce ([7]). E-shops and e-buyers are represented by shop and seller, and respectively client and buyer agents. In the experiments we considered simplified version of this scenario that involves a single shop agent S and n client agents C_i, $1 \leq i \leq n$. The shop agent is selling m products $\mathcal{P} = \{1, 2, \ldots, m\}$. We assume that each client agent C_i, $1 \leq i \leq n$, is seeking a set $\mathcal{P}_i \subseteq \mathcal{P}$ of products. Shop agent S is using m seller agents S_j, $1 \leq j \leq m$ and each seller agent S_j is responsible for selling a single product j. Each client agent C_i is using buyer agents B_{ik} to purchase products from the set $\mathcal{P}_i$. Each buyer agent B_{ik} is responsible with negotiating and buying exactly one product $k \in \mathcal{P}_i$, $1 \leq i \leq n$. To attempt purchase buyer agents B_{ik} migrate to the shop agent S and engage in negotiations; a buyer agent B_{ik}, that was spawned by client agent C_i, will engage in negotiation with seller S_k, to purchase product k. This simple scenario is sufficient for the purpose of our paper, i.e. to illustrate how a number of rule-based automated negotiations can be performed concurrently. In this setting, each seller agent S_j plays the role of a negotiation host defined in [3]. Therefore, in our system, we have exactly m instances of the framework described in [3]. Each instance is managing a separate ”negotiation locale,” while all instances are linked to the shop agent S. For each instance we have one separate set of rules together with a negotiation template that describes the negotiation mechanism implemented by that host. Note that each seller may use a different negotiation mechanism (different form of an auction, or an auction characterized by different parameters, such as the starting price or the bidding increment). See [2] for the details of our implementation of this conceptual architecture using JADE ([9]) and JESS ([10]).

English Auctions and Participants Strategy. For the purpose of this paper we have set our system up for a particular negotiation scenario involving English auctions. In an English auction there is a single item sold by a single seller and many buyers bidding against each other to buy this item. Usually, there is a time limit for ending the auction, a seller reservation price that must be met by the winning bid for the item to be sold and a minimum value of the bid increment. A new bid must be higher than the currently highest bid plus a minimal bid increment in order to be accepted. All the bids are visible to all auction participants.

The constraints describing English auctions were encoded as a modularized set of JESS rules. The rules were then used to initialize the negotiation hosts [2]).

Strategies of participant agents are defined in accordance with the negotiation protocol (i.e. English auction in this particular setting). Basically, strategy defines if and when a given participant will submit a proposal and what are its parameters. For the time being we opted for a simple solution: the participant submits a first bid immediately after it is granted admission and whenever it gets a notification that another participant issued an accepted proposal. The value of the bid is equal with the sum of the currently highest bid and an increment value that is private to the participant. Each participant has its own valuation of the negotiated product. If the value of the new bid exceeds this value then the proposal submission is canceled.

Strategies were implemented in Java as behaviors of participant JADE agent ([9]). In the future we plan to investigate and experiment also with rule-based representations of strategies ([6, 8]). This will provide us with the required flexibility and allow to easily add multiple strategies to our implementation.

Experiment. In the experiment we considered $m = 10$ products and $n = 12$ clients seeking all of them, i.e. $\mathcal{P}_i = \mathcal{P}$ for all $1 \leq i \leq 10$. Auction parameters were the same for all auctions: reservation price 50 and the minimum bid increment 5. Clients reservation prices were randomly selected from the interval [50,72] and their bid increments were randomly selected from the interval [7,17].

In this experiment 143 agents were created: 1 shop S, 10 sellers S_i, $1 \leq i \leq 10$, 12 clients C_i, $1 \leq i \leq 12$, and 120 buyers B_{ik}, $1 \leq i \leq 12$, $1 \leq k \leq 10$; and 10 English auctions were run concurrently. One separate JESS rule engine was also created for each English auction (therefore a total of 10 JESS rule engines were run in parallel). A total of more than 1400 messages was exchanged during negotiations. This means that the average number of messages exchanged per negotiation was about 140. While the total number of agents and messages is still small (for instance in comparison with these reported in [5]) these results indicate that the proposed approach has good potential for supporting large-scale experiments. Figure 1 shows messages exchanged between seller S_1 and buyers B_{i1}, $1 \leq i \leq 12$ that were captured with the JADE sniffer agent ([9]).

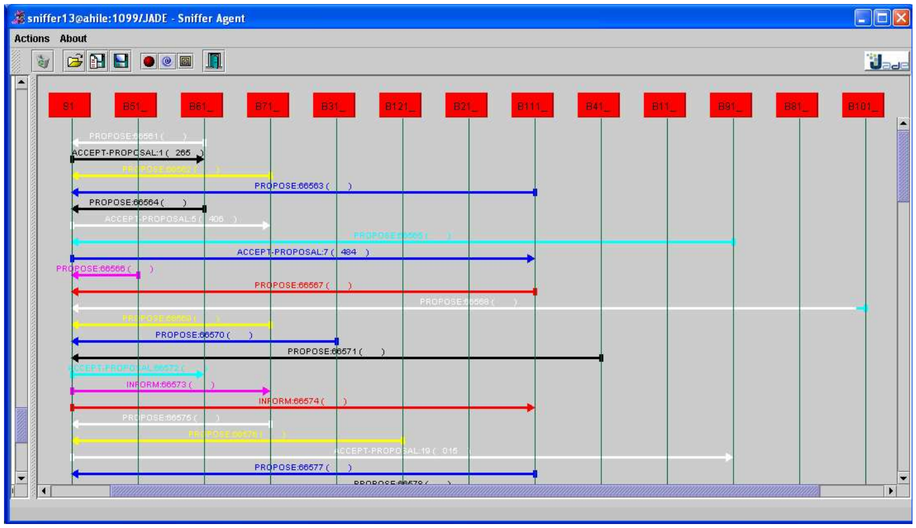

Fig. 1. Negotiation of a seller with 12 buyers in an English auction

Analyzing the results we have noticed that all auctions finished successfully. Let us comment on the result of auction for product 9. This auction finished with winner $B_{12,9}$ and winning price 66. Clients C_i ($1 \leq i \leq 12$) reservation prices (i.e. maximum prices clients would agree to pay for product 9) were: 71, 70, 70, 70, 70, 70, 70, 70, 50, 50, 70, 70 and their bid increments were: 10, 15, 8, 7, 9, 17, 17, 13, 11, 6, 16, 12. This means that when agent $B_{12,9}$ got its bid of 66 accepted, neither $B_{i,9}$ agent ($i \neq 12$) was able to bid more without exceeding its own reservation price. For example, $B_{4,9}$ would have to bid $66 + 7 = 73 > 70$ which was not acceptable according to $B_{4,9}$Šs strategy.

We have also noticed that many bids were rejected because the rule stating that ”a new bid must be higher than the currently highest bid plus a minimal bid increment” was violated. This might look incorrect because all buyers bid increments were set to values greater than 5 (the minimum value of the bid increment required by the auctions). The explanation is the race condition that occurs between competing buyer agents that bid against each other almost simultaneously. When they decide to bid, agents compute their bid correctly. However, when the first of the competing bids that reaches the host is validated, the currently highest bid is updated and this update might invalidate other incoming bids that didn’t reach the host on time.

4 Conclusions and Future Work

In this paper we discussed rule-based approaches to automated negotiations in multi-agent e-commerce systems. The discussion was supplemented by providing experimental results obtained using our own implementation of a rule-based price negotiation framework. These results support the claim that rules are a feasible and scalable technology for approaching flexible automated negotiations in e-commerce.

As future work we plan to: (i) complete the integration of the rule-based framework into our e-commerce model ([1]); (ii) asses the generality of our implementation by extending it to include other price negotiation mechanisms and multiple strategy modules. We will report on our progress in subsequent papers.

References

1. Bădică, C., Ganzha, M., Paprzycki, M.: UML Models of Agents in a Multi-Agent E-Commerce System. In: *Proc. ICEBE'2005*, Beijing, China. IEEE Computer Society Press, Los Alamitos, CA, (2005) 56–61.
2. Bădică, C., Bădiţă, A., Ganzha, M., Iordache, A., Paprzycki, M.: Rule-Based Framework for Automated Negotiation: Initial Implementation. In: A. Adi, S. Stoutenburg, S. Tabet (eds.): *Proc. RuleML'2005*, Galway, Ireland. LNCS 3791, Springer Verlag (2005) 193–198.
3. Bartolini, C., Preist, C., Jennings, N.R.: A Software Framework for Automated Negotiation. In: *Proc. SELMAS'2004*, LNCS 3390, Springer Verlag (2005) 213–235.
4. Benyoucef, M., Alj, H., Levy, K., Keller, R.K.: A Rule-Driven Approach for Defining the Behaviour of Negotiating Software Agents. In: J.Plaice et al. (eds.): *Proceedings of DCW'2002*, LNCS 2468, Springer verlag (2002) 165–181.
5. Chmiel, K., Tomiak, D., Gawinecki, M., Karczmarek, P., Szymczak, Paprzycki, M.: Testing the Efficiency of JADE Agent Platform. In: *Proc.* 3^{rd} *International Symposium on Parallel and Distributed Computing*, Cork, Ireland. IEEE Computer Society Press, Los Alamitos, CA, USA, (2004), 49–57.
6. Dumas, M., Governatori, G., ter Hofstede, A.H.M., Oaks, P.: A Formal Approach to Negotiating Agents Development. In: *Electronic Commerce Research and Applications*, Vol.1, Issue 2 Summer, Elsevier Science, (2002) 193–207.
7. Ganzha, M., Paprzycki, M., Pîrvănescu, A., Bădică, C., Abraham, A.: JADE-based Multi-Agent E-commerce Environment: Initial Implementation, In: *Analele Universităţii din Timişoara, Seria Matematică-Informatică*, Vol.XLII, (2004), 79–100.
8. Governatori, G., Dumas, M., ter Hofstede, A.H.M., and Oaks, P.: A formal approach to protocols and strategies for (legal) negotiation. In: Henry Prakken, editor, *Proc.* 8^{th} *International Conference on Artificial Intelligence and Law*, IAAIL, ACM Press, (2001) 168–177.
9. JADE: Java Agent Development Framework. See `http://jade.cselt.it`.
10. JESS: Java Expert System Shell. See `http://herzberg.ca.sandia.gov/jess/`.
11. Lochner, K.M., Wellman, M.P.: Rule-Based Specification of Auction Mechanisms. In: *Proc. AAMAS'04*, ACM Press, New York, USA, (2004).
12. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification scheme for negotiation in electronic commerce. In: F. Dignum, C. Sierra (Eds.): *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, LNCS 1991, Springer Verlag (2002) 19–33.
13. Skylogiannis, T., Antoniou, G., Bassiliades, N.: A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies – Preliminary Report. In: Boley, H., Antoniou, G. (eds): *Proc. RuleML'04*, Hiroshima, Japan. LNCS 3323 Springer-Verlag (2004) 205–213.
14. Tamma, V., Wooldridge, M., Dickinson, I: An Ontology Based Approach to Automated Negotiation. In: *Proceedings AMEC'02: Agent Mediated Electronic Commerce*, LNAI 2531, Springer-Verlag (2002) 219–237.
15. Wurman, P.R., Wellman, M.P., Walsh, W.E.: The Michigan Internet AcutionBot: A Configuarable Auction Server for Human and Software Agents. In: *Proceedings of the second international conference on Autonomous agents. Agents'98*, Minneapolis, USA. ACM Press, New York, USA, (1998), 301-308.
16. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A Parameterization of the Auction Design Space. In: *Games and Economic Behavior*, 35, Vol. 1/2 (2001), 271–303.